Deep Learning Project

Project on Siamese Convolutional Neural Network on Painter By Numbers dataset

Aya Mahagna

Contents:

The project goal:

The main goal of this project is to build a Siamese CNN (convolutional neural network) on Painter By Numbers dataset which detects if the paintings were drawn by the same artist or not.

Data:

We used several data files/directories to train and create our model:

1. all_data_info.csv file:

   Type: CSV file.

   Contains:

   - all the paintings' details by the artist
   - Genre: train/test

   Number of images: 103250 different images/paintings,

   Number of artists: 2319 different artists,

   The number of painting genres: 43.

   Note: an artist often paints in one genre, but we did not use it in our training since it isn't necessary for all artists, we used only the artist and the filename columns in the database.

2. Train and test files:

   Contains: all the images

   Note: there were corrupted images which caused many errors. We solved this issue by replacing these corrupted images using replacement_for_corrupted_files. Then we changed these images in train and test. Another issue we faced was the size of the files since the files were too big (causing many memory problems). We solved this issue by resizing the images to 256* 256 to get less running time (smaller size than 256 will affect the quality of the image and the training accuracy), a similar image size for all images, and better performance.

   There were paintings that their artists are unknown, so the artist was in the name of the painting and as a result there were artist names which were

the painting name/description and appeared only once, so we filtered the data to exclude the artists paintings which appeared only once or several times to get better training results and accuracy.

Architecture:

The architecture is a Siamese convolutional neural network (CNN) consisting of a shared network and a fully connected layer for classification.

The shared network is composed of four convolutional layers followed by ReLU activation and max pooling layers. The convolutional layers use a 3x3 kernel size and a stride of 1. The max pooling layers have a kernel size of 2 and a stride of 2. The output of the shared network is flattened and passed through two fully connected layers with ReLU activation.

The Siamese network has several identical copies of the shared network. Each copy of the shared network takes an input image and produces a feature vector. These feature vectors are concatenated and passed through another fully connected layer with ReLU activation. The output of the final fully connected layer is a feature vector that are used for classification.

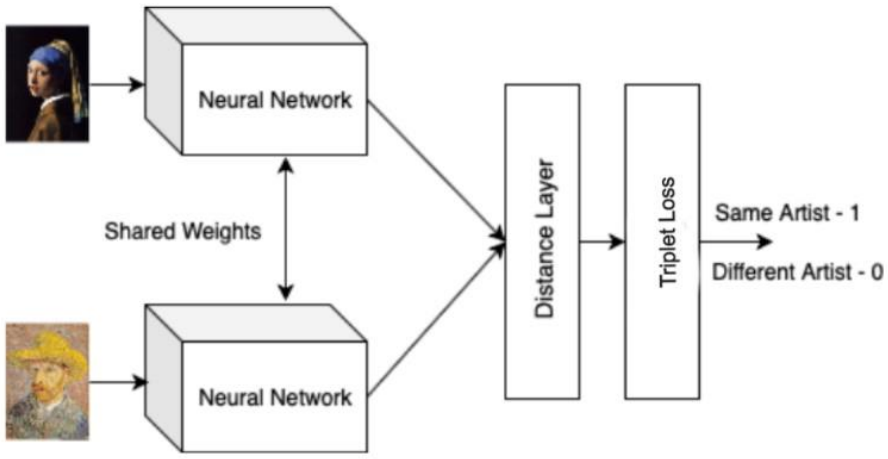


Figure 1. Siamese Neural Network Architecture

Testing and training:

In order to train Siamese CNN, we implemented several data loaders that were designed to support triplet loss, one for the test which prepares pairs of images and the distance to distinguish if they belong to the same artist, and the TripletDataSet which prepares triple images in the dataset the anchor and positive (belongs to the same artist) and the negative (belongs to different artist) and uses triplet loss function to calculate the loss.

The aim is to get to a point where the image is closest to the images drawn by the same artist, and farthest from those which are drawn by a different artist.

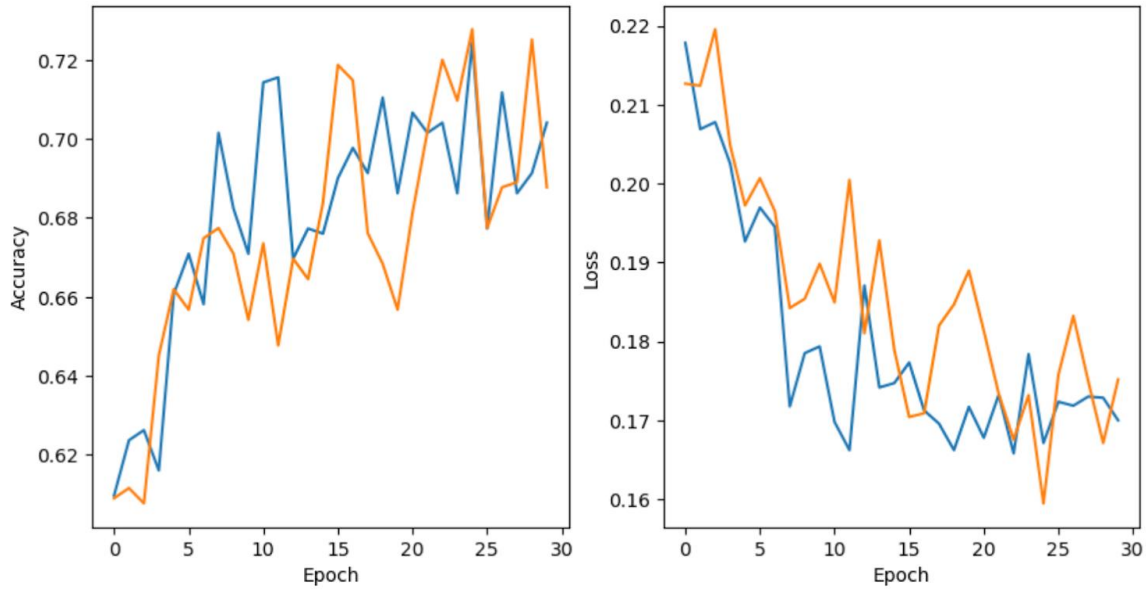
We also used optimizations (Adam, learning rate scheduler) to train the model and get better performance.

Some training loss and accuracy per epoch output:

```
Training:

epoch 1
Training Loss: 0.23

Steps: 0it [00:00, ?it/s]

Training success: 0.61

Steps: 0it [00:00, ?it/s]

Validation success: 0.61
epoch 2
Training Loss: 0.23

Steps: 0it [00:00, ?it/s]

Training success: 0.62

Steps: 0it [00:00, ?it/s]

Validation success: 0.61
epoch 3
Training Loss: 0.22

Steps: 0it [00:00, ?it/s]
```

Results:

These graphs are the results of our training, as we can see approximately the loss is being smaller every epoch and the accuracy/success is approximately getting better (there are many ups and downs).



blue - training, orange - validation

Accuracy of the test is between 56-61 after many tests and attempts we couldn't reach a better accuracy.

```
Evaluating

Steps: 0it [00:00, ?it/s]

Test success: 0.57
```

How to run the code:

We used Kaggle to run the code, added the Painter-by-numbers dataset to the notebook, and ran all the cells with the accelerator GPU P100, there were sometimes memory issues where we needed to clean the memory and run the cells again, so we recommend to resize the files first (using change_size function) and then clean the cache and memory and run the code again using the new resized files.

Resources that helped us:

- https://www.kaggle.com/code/spyrosrigas/20-painters-classification-with-cnns-and-svms/notebook
- https://towardsdatascience.com/triplet-loss-advanced-intro-49a07b7d8905?gi=6ddb767a5f3a
- https://github.com/LRNavin/painters_by_numbers
- https://www.kaggle.com/code/nevoit/siamese-neural-networks-one-shot-image-recognition/notebook